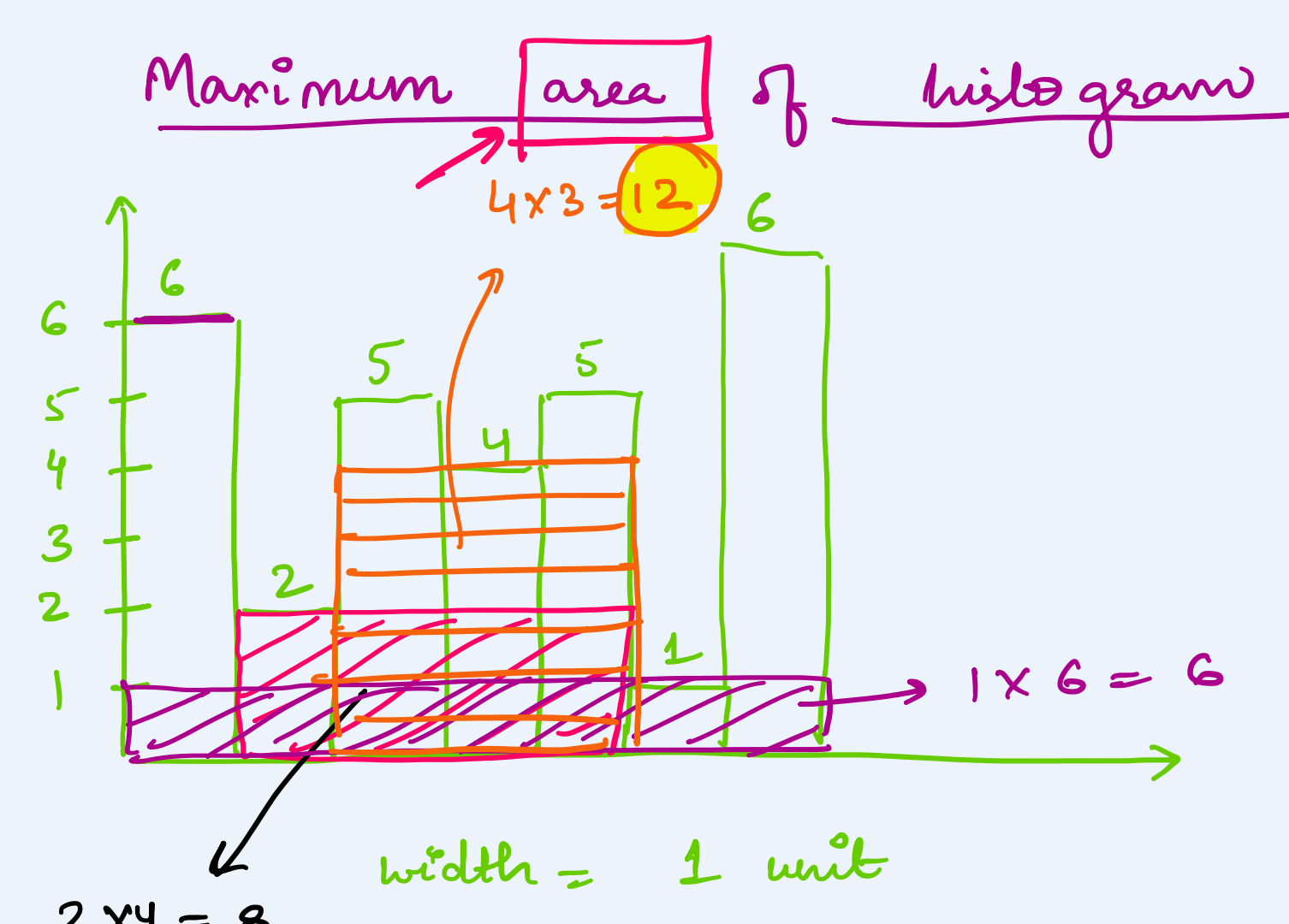
Sunday, 28 March 2021 3:49 PM

Q Maximum area of histogram



width = 1 unit

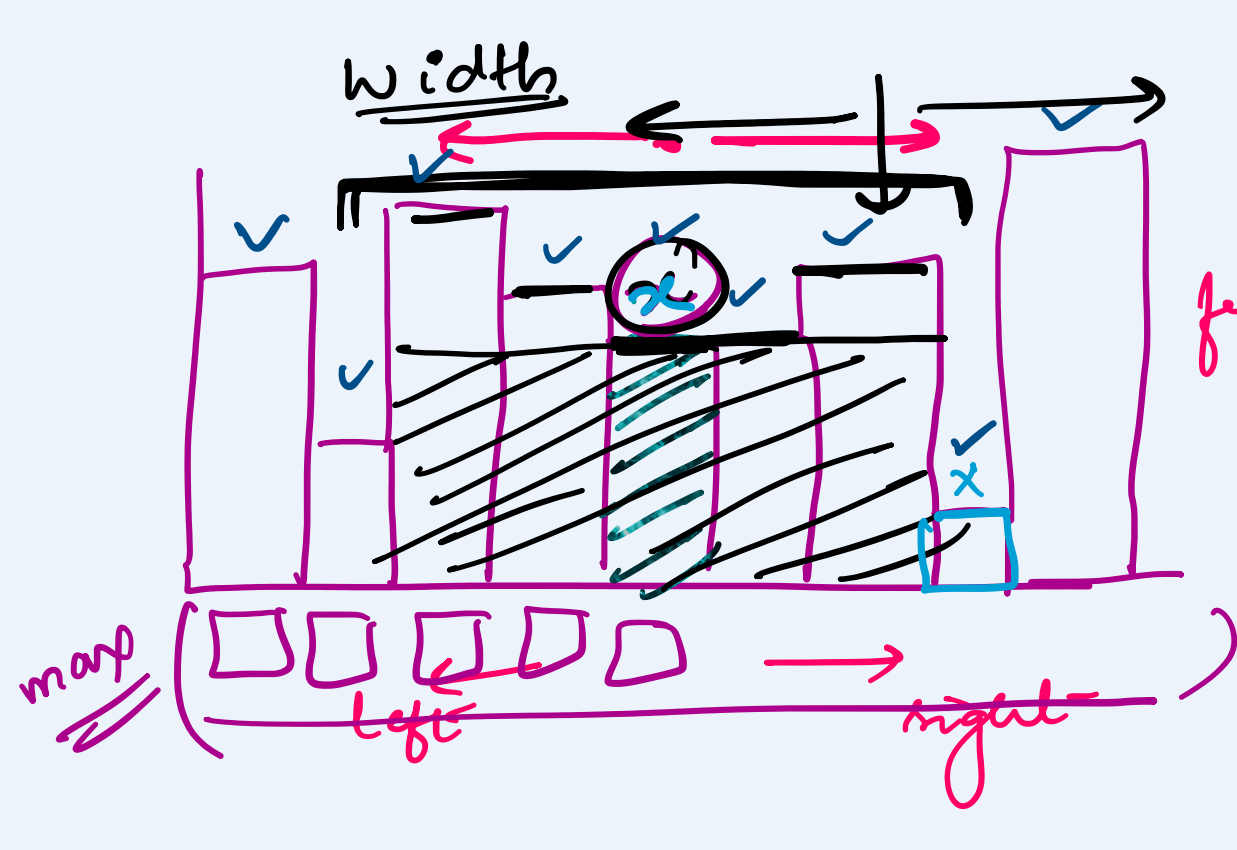
max (Area) → max (height) * max (width)

Height

* Height should atleast be equal to minimum bar.

Height will always be atleast one of the bars of histogram.

Width



width will come from both sides

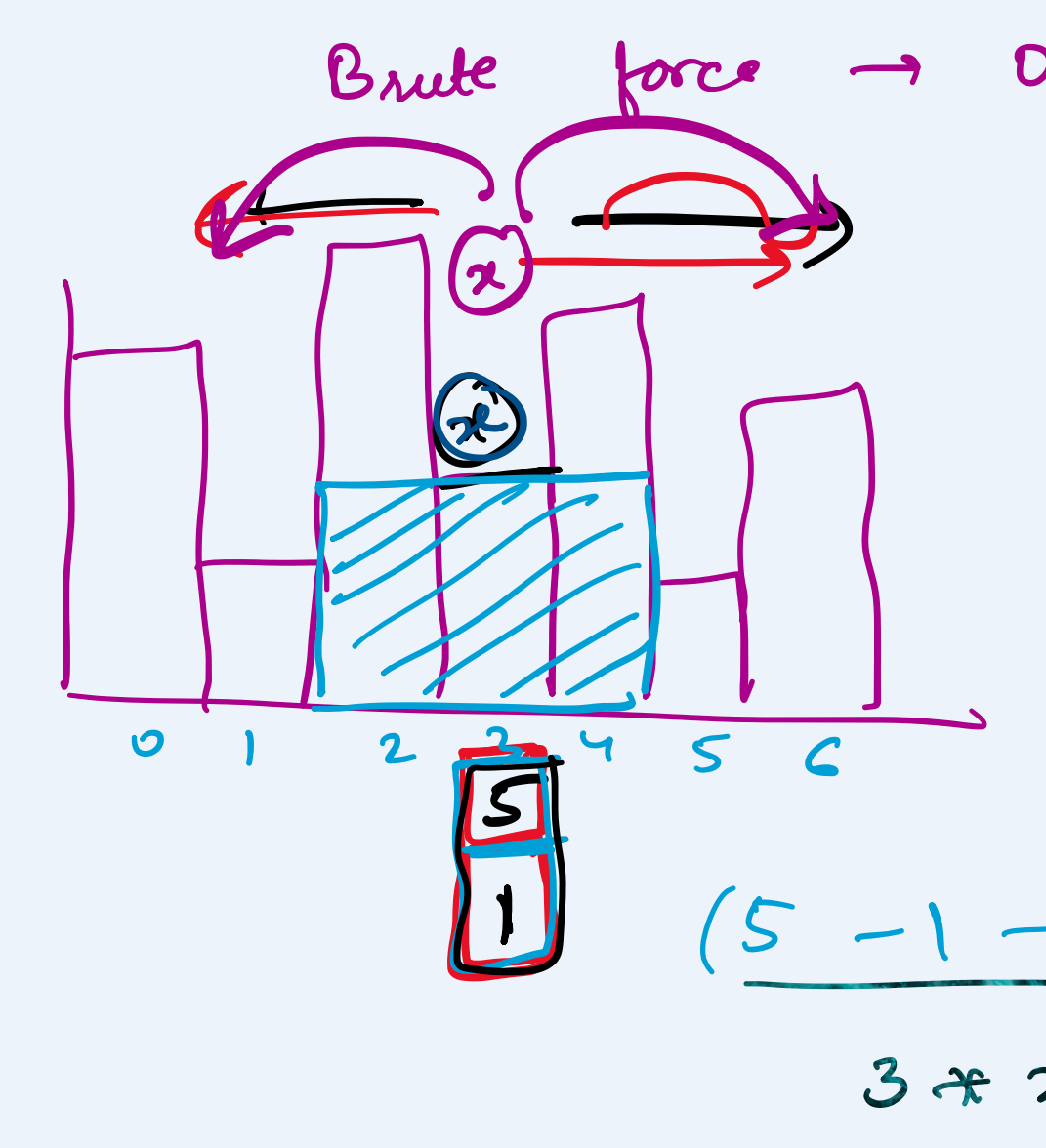
max (left → right)

Include bars only whose height >= current bar.

→ go to each and every bar, (height (x)) and see how much can be extended on both sides

→ 5 * x   4 + 1 = 5

Brute force → $O(n^2)$

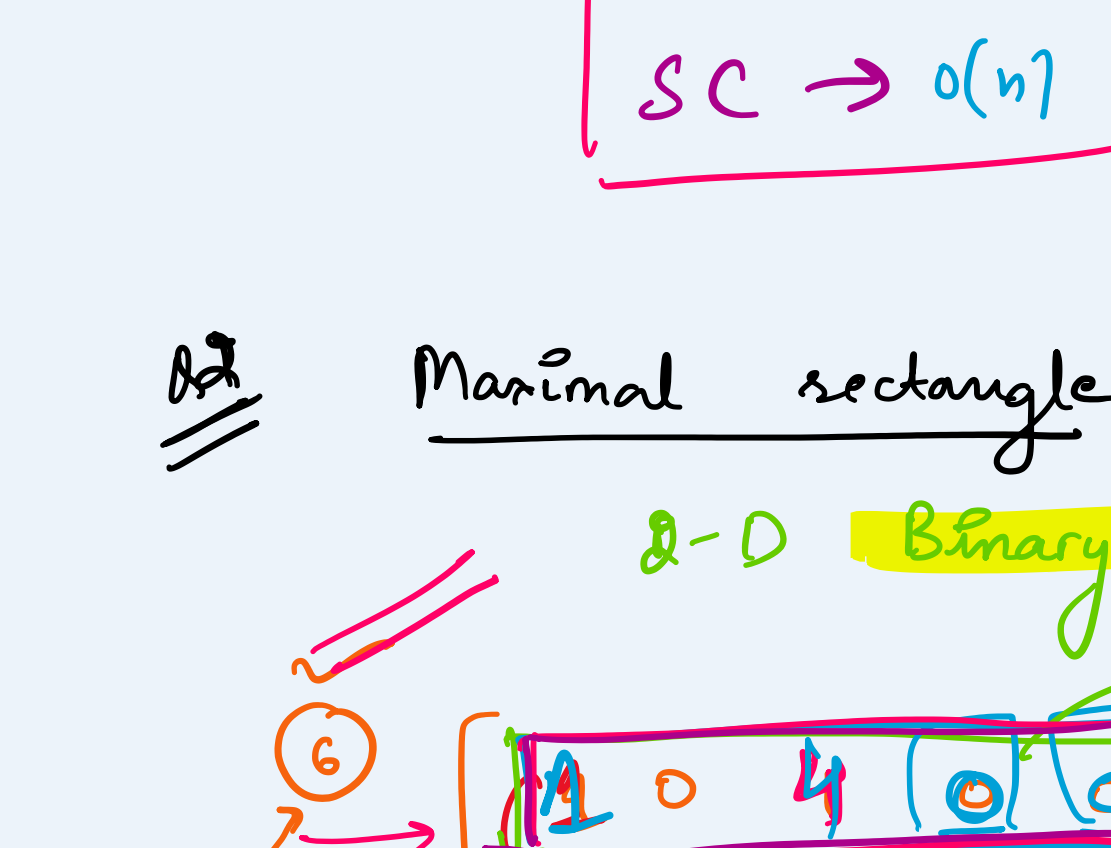


(5 − 1 − 1) * x = Area

3 * x = Area

Height of bars	6	2	5	4	5	1	6
	0	1	2	3	4	5	6
next smaller	1	5	3	5	5	7	7
prev smaller	-1	-1	1	1	3	-1	5
	6	10					

12



(6) 1 * (7 + 1 − 1) = 7
(7) 6 * (7 − 5 − 1) = 6

height * (next − prev − 1)

1. 6 * (1 + 1 − 1) = 6
2. 2 * (5 + 1 − 1) = 10
3. 5 * (3 − 1 − 1) = 5
4. 4 * (5 − 1 − 1) = 12
5. 5 * (5 − 3 − 1) = 5

1. next smaller
2. prev smaller } → O(n)
3.

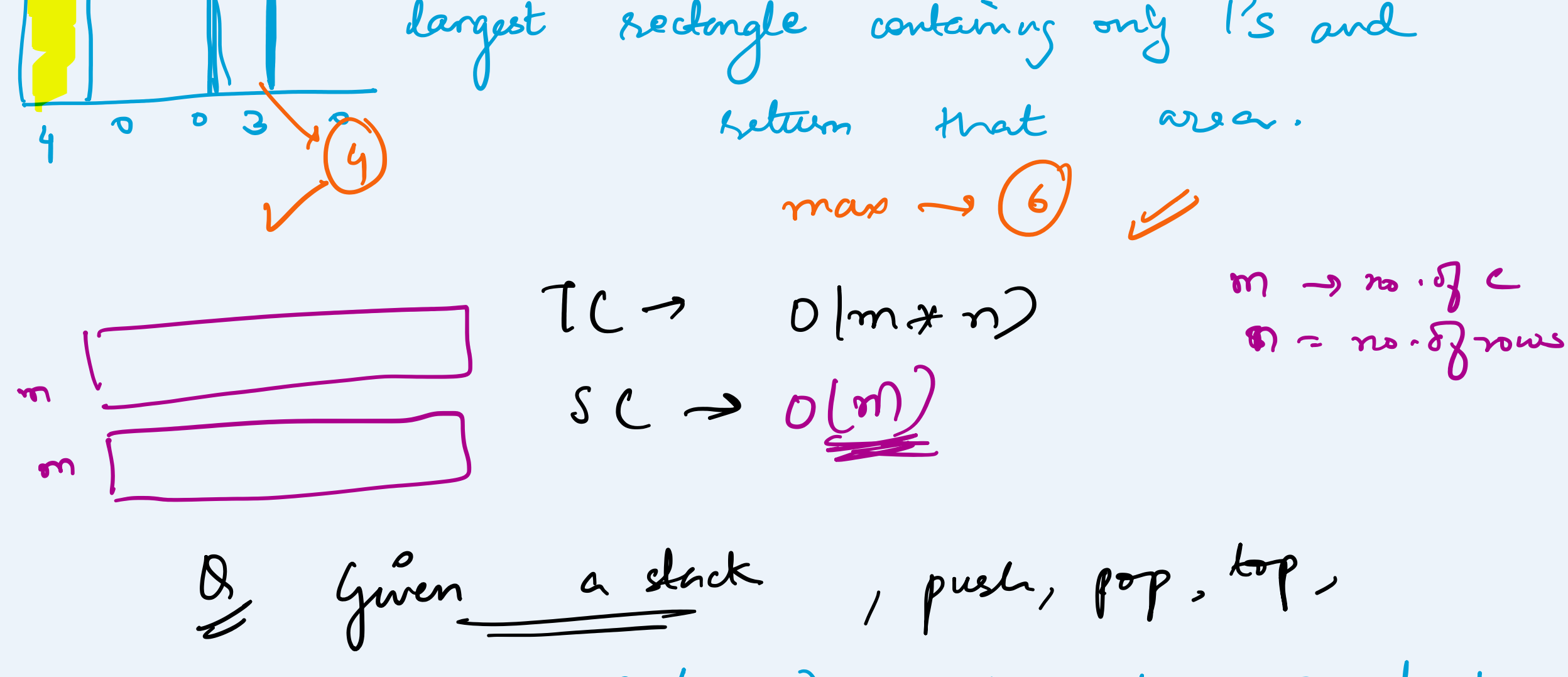
TC → O(n)
SC → O(n)

Q2 Maximal rectangle

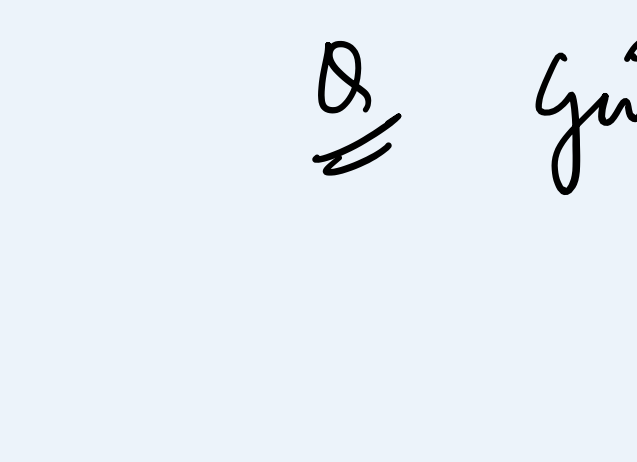
2-D Binary matrix

1	2	3
1	1	1

m * n → T(m×n)



largest rectangle containing only 1's and return that area.



max → 6

TC → O(m * n)
SC → O(m)

m → no. of c
n = no. of rows

Q Given a stack, push, pop, top,

getMin( ) min value of stack

5
3

stack | 12 | 22 | 6 | 3 | 3 |

min stack | 12 | 6 | 3 | 3 |

∞

12 < ∞

22 > 12

3 < 6

12

TC → O(1)
SC → O(n)

s.push(12);
22
s.getMin();
ms.peek();
s.push(3);
s.pop();
s.pop();

Approach

| 12 | 6 | 13 |

min = ∞ 12 6

TC → O(1)
SC → O(1)

| 12 | 6 | 13 | x − min |

x − min

new ele to be added x

min = 6 x

x < min

x − min < 0

x − min → −ve

1 − 2

x − min

| 2 | 3 | 4 | x − min | −1 |

x = 1

min = 2 1 2 1

−ve

s.pop()

−1

−ve

min = min − st.top
= 1 − (−1) = 2

Issue : −ve negative values

| −3 | −4 | −5 | −6 | −7 | −8 |

| 12 | 3 | 6 |

x

min = —

x < min
→ x − min < 0
Adding x on both sides
x + x − min < x
2x − min < x ← all values

x − min < x

If x is the new min,

⇒ x is less than all other values of stack

2x − min is lesser than x

2x − min in the stack

2x − min

| 12 | 3 | 6 | 2x − min 2(2) − 3 = 1 |

2

2 * 15 − 14 = 16
30 − 14
2x − min
30 − 16
−62

16, 1 −23

min = 3

pop

getmin

15

2(−23) − 16
−46 − 16

pop()

min = 16 −23

| 16 | 1 |

46
16
62

min = 16 15 14 15

2(14) − 15
28 − 15
13

pop

2x − min
2(14) − min

14 < 15 getMin → 14

pop()

s.pop()

2 * min − st.top
2 * 14 − 13
28 − 13

15

s.pop() < min

2 * (−23) − (−62)

16

x − min